

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of the claims:

1. (previously presented) A computer implemented method for managing a collaborative process that involves at least a first player in a first enterprise having a first collaborative process manager and a second player in a second enterprises having a second collaborative process manager comprising the steps of:

a) defining an inter-enterprise collaborative business process including templates and a plurality of work nodes; wherein each work node has a task role identifier for specifying one of the first player and the second player as responsible for execution of the work node, and the templates include definitions and a sharing scope that is one of public and process role specific;

b) the first collaborative process manager executing a first peer instance of the collaborative business process;

c) the second collaborative process manager executing a second peer instance of the collaborative business process; and

d) specifying the sharing scope of at least one template to keep data private between the first and second collaborative process managers;

wherein the first peer instance of the collaborative business process and the second peer instance of the collaborative business process form a logical execution instance; and

wherein the first peer instance of the collaborative business process and the second peer instance of the collaborative business process communicate through messages for information exchange and synchronization.

2. (previously presented) The method of claim 1 wherein the collaborative business process includes a plurality of tasks, the method further comprising the steps of:

e) the first collaborative process manager receiving a current task;

f) the first collaborative process manager determining if the current task is the responsibility of the first collaborative process manager;

- g) when the current task is the responsibility of the first collaborative process manager, executing the current task; and
- h) when the current task is not the responsibility of the first collaborative process manager, not executing the current task.

3. (previously presented) The method of claim 2 wherein the step of when the current task is the responsibility of the first collaborative process manager, executing the current task further comprises the steps of:

- g_1) scheduling the current task;
- g_2) dispatching the current task for execution;
- g_3) when the execution is complete, generating a task return message; and
- g_4) sending the task return message to the second collaborative process manager.

4. (previously presented) The method of claim 2 wherein the step of when the current task is not the responsibility of the first collaborative process manager, not executing the current task further comprises the steps of:

- h_1) not executing the current task;
- h_2) waiting for a task return message from the second collaborative process manager; and
- h_3) receiving a task return message from the second collaborative process manager.

5. (previously presented) The method of claim 4 wherein the step of when the current task is not the responsibility of the first collaborative process manager, not executing the current task further comprises the steps of:

- h_4) evaluating the current task return message to determine whether an out-of order condition has occurred;
- h_5) when an out-of-order condition has occurred, queuing the task return message for later processing; and

h _6) an out-of-order condition has not occurred, processing the next task by employing the task return message.

6. (original) The method of claim 1 further comprising:

using a cooperation key to identify a logical instance of the collaborative business process and to correlate and synchronize multiple peer instances of the execution of a single collaborative business process.

7. (original) The method of claim 4 employing task return messages for synchronizing the peer process instances and for exchanging data between the process instances; wherein each task return message includes

- a cooperation key for specifying a logical process instance;
- a local handle of the process instance and task;
- an activity execution status; and
- a sub-packet of process data passed to a task.

8. (original) The method of claim 1 wherein the collaborative process has a list of process-roles for indicating logical participants of the collaborative process; wherein each work node has a task role that matches one of the process roles; and wherein a peer process having a process role that matches the task role of a work node is responsible for executing the work node.

9. (original) The method of claim 1 further comprising the step of:

providing a collaborative process definition language (CPDL) for use in defining collaborative business processes.

10. (canceled)

11. (currently amended) The method of claim 1 wherein the step of specifying the sharing scope of at least one template includes:

setting the sharing scope as public; wherein a~~the~~ data object is public to all process-roles.

12. (currently amended) The method of claim 1 wherein specifying the sharing scope includes the step of

setting the sharing scope as process-role specific for a particular process role; wherein a~~the~~ data object is accessible only to the process-role specified.

13. (currently amended) The method of claim 1 wherein specifying the sharing scope includes the step of

setting the sharing scope as process-role specific for at least two different process roles; wherein a~~the~~ data object is accessible only to the specified process roles.

14. (previously presented) A computer system for allowing a first player in a first enterprise to collaborate with a second player in a second enterprise comprising:

a collaborative business process definition specified by a collaborative process definition language and based on an inter-enterprise business collaboration protocol, the collaborative business process definition having a plurality of work nodes, each work node having a task role;

a first collaborative process manager in the first enterprise for executing a first peer process instance of the collaborative business process definition, the first peer process instance having a role; wherein the first peer process instance is responsible only for the work nodes that have a role that matches the role of the first peer instance;

a second collaborative process manager in the second enterprise for executing a second peer process instance of the collaborative business process definition, the second peer process instance having a role; wherein the second peer process instance is responsible only for the work nodes that have a role that matches the role of the second peer instance;

an out-of-order handler mechanism for receiving messages from other collaborative process managers, determining whether messages are received out of order,

when messages are received out of order, halting execution, and when messages are not received out of order, continuing with execution;

wherein the first peer process instance and the second peer process instance form a single logical execution instance; wherein the logical execution instance is identified by a cooperation key that is assigned to the first peer process instance and the second peer process instance; and

a peer to peer communication mechanism for enabling data exchange and synchronization between the first peer process instance and the second peer process instance.

15. (previously presented) The system of claim 1 further comprising:

a task return message generator for generating a task return message for each task executed by the collaborative process manager.

16. (canceled)

17. (previously presented) The system of claim 14 further comprising:

an private sub-process manager for selectively making process data objects private to a particular collaborative process manager.

18. (previously presented) The system of claim 14 further comprising:

a task role determination module for receiving the current task, for determining whether the current task is the responsibility of the collaborative process manager, when the current task is the responsibility of the collaborative process manager, for scheduling and dispatching the task for execution, when the current task is not the responsibility of the collaborative process manager, not executing the current task.

19. (original) The system of claim 15 wherein each task return message includes

a cooperation key for specifying a logical process instance;
a local handle of the process instance and task;
an activity execution status; and

a sub-packet of process data passed to a task.

20. (original) The system of claim 14 wherein the collaborative business process definition includes a list of process-roles for indicating logical participants of the collaborative process; wherein each work node has a task role that matches one of the process roles; and wherein a peer process having a process role that matches the task role of a work node is responsible for executing the work node.